We can do this by trading off bandwidth -- which we have in abundance -- for scarce power. This tradeoff comes from Shannon's famous channel capacity formula:

$$C = B * \log_2(1+S/N)$$

where C is the channel capacity in bits/sec. B is the channel bandwidth. S is signal power and N is noise power. As long as we do not try to exceed the capacity C of a channel, it is theoretically possible to communicate with an arbitrarily low error rate. Above C, it is simply not possible to do so

Note the relationship between B and S/N: to a certain extent we can compensate for a lower S/N by *increasing* B. Now this may be counterintuitive to many of you; after all, it's standard practice to use the narrowest available filter when copying an especially weak CW signal. But it is very much true *provided that we carefully design a wideband signal and build a matched receiver filter.*

Since noise power is directly proportional to bandwidth, Shannon's formula can be rewritten as

$$C = B * \log_2(1+S/N_0B)$$

where N_0 is the noise spectral density in watts per hertz. So as B goes to infinity we reach a point of diminishing returns. But more bandwidth is always beneficial, even though the incremental benefit may be small.

A way to rewrite Shannon's law that better expresses the tradeoff between bandwidth and power is as follows:

$$E_b/N_0 > B/R * (2^{R/B} - 1)$$

Here we've introduced two new variables: R, the actual signalling rate in bits/sec and Eb, the energy per user data bit measured in watt-seconds (joules). No is again the noise spectral density in W/Hz, but this too has units of joules. This formula says that the minimum required Eb/No ratio is a direct function of the available bandwidth relative to the data rate. The more bandwidth, the lower the necessary E_b/N_0 ratio and vice versa. But the relationship is nonlinear; even with infinite B/R, Eb/No must always be greater than ln(2). Expressed in decibels, this is -1.6dB -- the famous Shannon Limit.

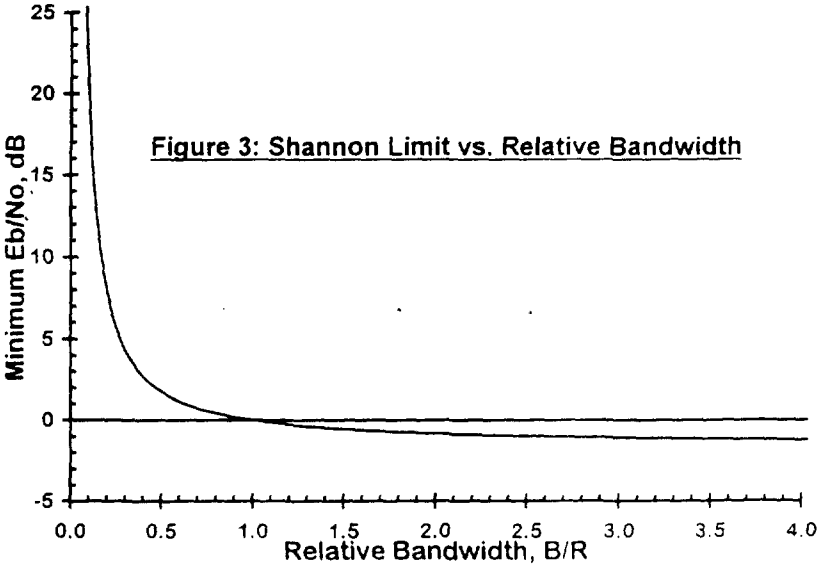

Figure 3: Shannon Limit vs. Relative Bandwidth

The ratio Eb/No is an absolutely fundamental one in digital communications, as it directly expresses the power efficiency of a coding and modulation system. That is, it concentrates on what really counts: how much RF energy does it take to send each bit of user data over a channel with a certain noise spectral density?

b. Fading:

Another important characteristic of the EME channel is that it fades. The signal takes many slightly different paths as it reflects off the uneven, slowly librating lunar surface. It can also split into multiple paths as it passes through the earth's ionosphere or troposphere. Because these paths vary in relative length, the various signal components arrive at the receiver at slightly different times and with different phases. The receiver "sees" the vector sum of these components. Since their relative path lengths change with time as the moon librates, sometimes they add to produce an enhanced signal. At other times they cancel, producing a degraded signal. At low antenna elevation angles, multipath reflections (ground reflection gain) from the surface of the earth are also important.

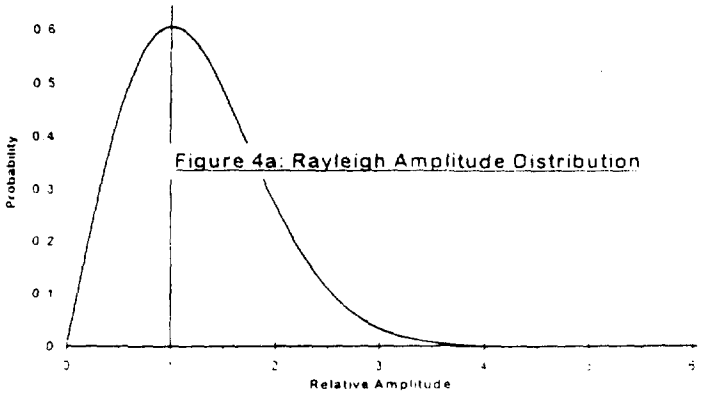

Figure 4a: Rayleigh Amplitude Distribution

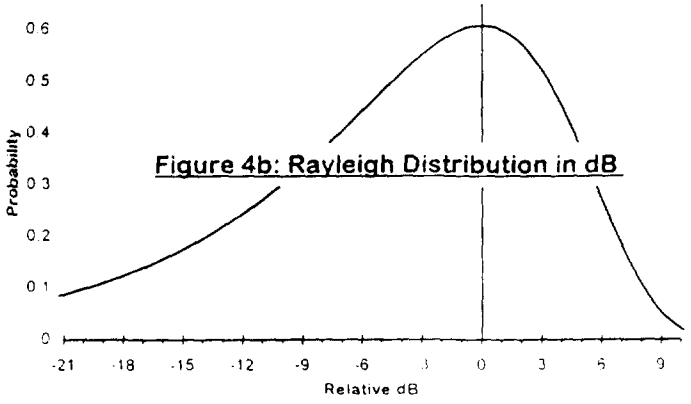

Figure 4b: Rayleigh Distribution in dB

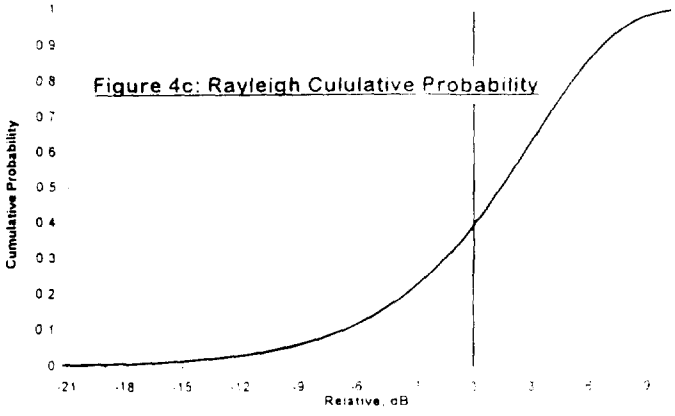

Figure 4c: Rayleigh Cululative Probability

It turns out that the precise physical mechanisms that produce multipath fading aren't really that important. Somewhat different mechanisms produce the multipath fading seen in land mobile radio, but it resembles EME in that many separate paths are often present. And when there are many reflection paths without any direct line-of-sight paths, we have a "Rayleigh fading channel". That is, the signal amplitude distribution follows a Rayleigh probability function and the signal phase is uniformly distributed at all angles.

This doesn't quite tell the whole story. If you sample a fading signal, wait a very short time and sample it again, the two measurements will tend to have similar amplitude and phase, i.e., they are correlated. If you wait a long time between measurements, they will tend to be different, i.e., uncorrelated. The time scale beyond which a pair of signal measurements tend to become uncorrelated is the "channel coherence time". It depends on the relative physical velocities of the various objects involved in reflecting or refracting the RF energy. It also depends on RF wavelength. Faster velocities and shorter RF wavelengths produce shorter coherence times, i.e., "faster" fading. For example, with all other things constant, mobile FM "flutter" and EME fading are both typically 3x faster on 70cm than on 2m. As an extreme example, aurora scatter has extremely short coherence times because of the very high apparent "velocities" of the ionized cloud particles.

Coherence times on 2m EME are typically several seconds, but this can vary over the month and with ionospheric or tropospheric activity. Because the multipath components arriving at the receiver travel many different distances, their relative phase (and whether they add or cancel each other) depends on the precise wavelength. A small change in frequency may be enough to cause two paths that formerly canceled to enhance each other, or vice versa. The frequency scale over which this occurs is called the "coherence bandwidth".

Coherence time depends on the relative velocities of different parts of the propagation medium with respect to the observer: e.g., lunar libration, which varies with time. On the other hand, coherence bandwidth depends on the size of the propagation medium: the larger the medium or object, the wider the possible range of multipath delays that can occur. Longer delay differences between multipath components mean that smaller frequency differences are enough to comprise a significant fraction of a wavelength that can turn a peak into a null and hence the narrower the coherence bandwidth.

As previously discussed, the maximum EME delay spread possible given the moon's diameter is 11.6 milliseconds, corresponding to a coherence bandwidth of about 86 Hz. If the lunar surface reflected RF the way it reflects light, this would indeed be the coherence bandwidth; but in fact it's more like 500-1000 Hz on the lower EME bands.

Why the discrepancy? Because the lunar surface reflects RF differently than it does light. At optical wavelengths, the full moon appears uniformly bright (discounting albedo variations between maria and highlands) because, thanks to the fine layer of dust covering its entire surface, the lunar surface is extremely rough at scales corresponding to visible wavelengths. And the moon's surface is fairly rough at large scale (much longer than VHF/UHF wavelengths) due to mountains, craters and the like. But in between, the moon is relatively smooth over a fairly wide range of radio wavelengths. So instead of uniformly scattering RF at all angles, the lunar surface tends to reflect it more specularly; the center of the disk appears brighter than the limb. But the limb isn't totally dark because of the large scale surface undulations. So the moon can be best seen as a collection of many independent specular reflectors, something like a dance hall mirror ball where the glue holding the mirrors on has softened and allowed the mirrors to shift their angles with respect to the surface.

So the net result is that the delay spread is somewhat shorter (and the coherence bandwidth wider) than it would be if the moon scattered RF as uniformly as it does light. The coherence time is somewhat longer too, thanks to the smaller contribution from the more rapidly moving limb. On the other hand, the smaller reflecting area means a higher average round trip path loss.

In the discussion that follows, it's important to remember that fading affects signal phase as well as amplitude. It is typical to see very rapid phase shifts (approaching 180°) across a deep fade. This is easiest to understand by example. Consider a path with just two multipath components of nearly equal amplitude and 180° out of phase. If the first component starts at a lower amplitude than the second component but then increases to where it becomes the stronger, it's easy to see how the amplitude of the vector sum of both components goes through zero and how the phase will suddenly jump 180° during the transition.

c. Coherent vs. noncoherent (sometimes called incoherent) demodulation:

A receiver always works best when it has a signal carrier phase reference. With it, noise in phase quadrature to the desired signal can be ignored. But a receiver without a phase reference must respond to all incoming signal phases. The benefit is slight at high SNR (where the signal "swamps" the noise) but can be large at low SNR. This is why AM envelope detectors have a threshold effect on weak signals while SSB receivers do not. (Noncoherent FM demodulators have an even more pronounced threshold effect.)

[Note from KA9Q: see diagram of signal vector plus noise for large and small signal case.]

[Note from W3IWI: Phil and I were working on this paper right up to the last minute and it wasn't possible for us to get the figures merged with the text. Sorry!]

Some digital modulation schemes require the receiver to maintain a stable carrier phase reference over relatively many data bits. A good example is binary phase shift keying (BPSK), used for telemetry on the AMSAT Phase III satellites and on the low-earth-orbiting PACSATs. Since the BPSK signal suppresses

the actual carrier, the receiver must reconstruct a local carrier reference from the data sidebands. (BPSK is just suppressed carrier DSB-AM with digital modulation.)

Two essentially equivalent methods are commonly used to generate a BPSK carrier reference: the Costas loop and the squaring loop. Both use a nonlinearity to regenerate the suppressed carrier: an I-Q multiplier in the Costas loop or a signal squarer in the squaring loop. This necessarily degrades the SNR of the recovered carrier by 6dB for binary PSK, 12 dB for QPSK and more for higher orders. This is generally not a problem when signals are strong, the channel is stable and relative frequency uncertainty is small, as in high speed line-of-sight links. One simply narrows the loop filter enough to raise the recovered carrier SNR to the desired level. I.e., the loop "smooths" its carrier phase estimate over many data bits. This works well as long as the channel coherence time (as well as other sources of phase noise, such as local oscillators) is long with respect to the reciprocal of the loop filter bandwidth. In a typical system designed for a relatively high SNR, the loop filter bandwidth might correspond to 30 bit times.

[Note: see diagram of costas loop with loop filter highlighted]

Things get difficult when strong FEC is used to get the most out of the limited signal power. At such low "raw" SNRs (or more precisely, Es/N0, the ratio of energy per channel symbol to the noise spectral density) an additional degradation called "squaring loss" appears. To compensate, the loop filter must estimate carrier phase over an even longer period. For example, KA9Q's 1200 bps experimental QPSK satellite modem estimates carrier phase over several hundred bit times, and yet the modem's overall performance is still limited by the noise that gets through this very narrow filter.

Things only get worse as we scale to lower signal powers and data rates. At 1 bps, several hundred bit times is several hundred seconds. This corresponds to a carrier loop filter bandwidth of less than .01 Hz! Even if the frequency instabilities in an amateur EME system could be kept below this level, we are faced by the insurmountable fact that the EME channel simply won't stay still for more than a few seconds -- the fading will "modulate" the signal right out of the carrier filter bandwidth. So it's pretty clear by now that conventional suppressed carrier BPSK with coherent demodulation is probably not a good choice for EME.

d. Coherent demodulation with a pilot carrier:

One way to mitigate the fading problem with BPSK is with a "pilot". That is, you put some fraction (e.g., 10%) of the total RF power into a residual unmodulated carrier (the pilot) for use by the demodulator in regenerating a local reference (e.g., with a narrow bandpass filter or PLL). You do this by phase modulating the carrier with less than +/-90 degrees of phase shift; the resulting signal is equivalent to a suppressed carrier BPSK signal plus an unmodulated carrier in phase quadrature.

[Notes: see diagram of signal vectors] [see diagram of loop with simple PLL tuned to carrier]

Because this carrier component can be recovered without a nonlinearity, there are no losses. Of course, the pilot is only part of the total signal power, and this power can't carry data. So there is an optimum pilot power fraction that depends on the situation.

Pilots are sometimes used even on channels that nominally don't fade. NASA has long used them on deep space links where signals are weak, data rates are low and relative frequency uncertainties are so high that the inherent losses in regenerating a suppressed carrier are just too great. Pilots also help deal with the small amounts of multipath fading that occurs when signals have to pass through planetary atmospheres or through the solar corona.

e. Noncoherent demodulation (especially m-ary FSK):

Many other alternatives exist when the receiver can't maintain a stable, long term carrier phase reference

BPSK can be differentially detected by simply multiplying the previous symbol by the current one. In essence using just the previous symbol as a carrier reference. Unfortunately this doesn't perform nearly as well as coherently demodulated BPSK, especially at low SNR, because only one bit's worth of energy is used as the reference.

[Note: see diagram of differential BPSK demod]

The most popular noncoherently demodulated signaling method is binary frequency shift keying (FSK). The demodulator simply measures the total signal energy -- ignoring phase -- over the bit time for each of the two "tones" and picks the one with the greater energy as its decision. This too has worse weak-signal performance than coherent BPSK.

But it turns out that frequency shift keying -- even with noncoherent detection -- can be made more power efficient by simply adding tones. Consider 4-ary FSK - i.e., FSK with four "tones". Only one tone is sent at a time, each representing log2(4) = 2 user data bits.

The 4-ary FSK demodulator is just an extension of the classic binary (2-ary) FSK demod. It has 4 filters and envelope detectors, one for each tone. At the end of each tone pulse (called a "symbol") the demodulator picks the filter channel with the greatest accumulated energy and outputs the 2 data bits that correspond to that symbol.

[Note: see diagram of 4-ary FSK demod]

Why does this scheme use less power per user data bit than binary FSK? Because each symbol represents 2 user data bits, we can afford to invest twice as much RF energy in each channel symbol as in binary FSK while maintaining the same energy per user data bit (Eb). Raising the symbol energy reduces the chance that a random noise peak in the three "incorrect" channels will cause it to be chosen over the correct channel. At reasonably low bit error rates, the improvement is dramatic; 4-ary FSK requires an Eb/N0 of 11.5 dB to maintain a 10^{-5} BER, and that's nearly 3dB than binary FSK. 8-ary FSK requires 9.5 dB, and that's slightly less than perfect BPSK!

But as the number of tone channels M increases, the improvement slows down. The probability that any one tone channel will cause an error is unchanged; it's just there are so many more of them and just one can cause an error. Also, the symbol energy increases only as $\log_2(M)$, so it falls further behind M.

[Note: see diagram of BER vs Eb/N0 curves vs M]

The bottom line is that as the number of channels M approaches infinity, M-ary FSK approaches an Eb/N0 of -1.6 dB. This is exactly the famous Shannon limit for infinite bandwidth. In fact, this was essentially the system that Shannon analyzed in his landmark 1948 paper on noisy channel capacity.

In a real system, of course, we have to pick a finite M and layer additional forward error correction coding on top of it. How large can we make M? There are two limits: complexity and bandwidth.

Complexity is no longer a practical problem. The British Foreign Office "Piccolo" system of the early 1960s used 32-ary FSK with each symbol representing one of the 32 letters in the Baudot alphabet. The Piccolo demodulator consisted of 32 separate hardware filters, one for each tone. That's obviously rather cumbersome.

Today we can use the Fast Fourier Transform (FFT) to build the receiver filter bank for just about any value of M we like. FFTs with thousands of points (bins) are no sweat on modern microcomputers. We just take the FFT of the received symbol in the time domain, square and add the real and imaginary components of each frequency "bin", and pick the bin with the largest energy.

The other limit is the bandwidth needed for all the tones. As M increases, the tone signalling rate decreases as $\log_2(M)$ because that's how many data bits are represented in each tone. Since the tones must be at least $1/T$ Hz apart, where T is the symbol time in seconds, we can space the tones somewhat more closely together as we increase M thus partially compensating for the extra tones. So the overall bandwidth requirements increase as $M/\log_2(M)$.

[Note: see frequency diagram of tone spacing]

It's interesting to note that the relative bandwidth required for 4-ary FSK is $4/\log_2(4) = 2$, the same as 2-ary FSK: $2/\log_2(2) = 2$. In other words, "the first one is free", much like the transition from binary PSK to QPSK (4-phase PSK). (QPSK can carry twice the data in the same bandwidth as BPSK with the same Eb/N0. But higher order PSK constellations require greater Eb/N0 for the privilege of further increasing the "bandwidth efficiency", just as higher order FSK constellations require greater bandwidth to save power.)

Some final comments. M-ary FSK is just one example of what is more generally known as a "M-ary orthogonal signal set". A set of properly spaced tones (sine functions) is just one example; others include Walsh functions (the Qualcomm CDMA digital cellular system uses 64-ary Walsh code modulation on its mobile-to-base link) and the pulse-position modulation (PPM) codes often used on optical fiber. The theoretical performance is exactly the same for all these schemes, but sine waves (i.e., FSK) have the practical advantage of maintaining orthogonality without precise time synchronization.

Also, the term "noncoherent detection" is slightly inaccurate here, even though it's standard in all the textbooks. It would be more accurate to say that detection is "noncoherent from symbol to symbol". The tone filter/detector is still very sensitive to tone phase *during* a symbol interval. For example, an incoming tone that suddenly phase flipped 180 deg degrees exactly halfway during the symbol interval would produce a zero output. To minimize this effect, the the signalling interval must be kept short relative to the coherence time of the channel. This creates another practical limit on M.

Here's another way to look at it: channel fading AM modulates the signal, and the receiver tone filters must be wide enough to capture the sidebands so generated. That puts a minimum bandwidth limit on the filter, which corresponds to a maximum coherent integration time.

<u>f. FEC with coherent and noncoherent modulation:</u>

The "rate" of a forward error correction (FEC) code is the ratio of the user data bit rate to the encoded channel symbol rate. For example, a convolutional code that produces two encoded symbols for every user data bit would be a rate 1/2 code, as would a block code that produces 24 encoded symbols for every 12 user data bits.

Many FEC schemes exist. Some, like Reed-Solomon (RS) block codes, are well suited to M-ary orthogonal modulation because they use non-binary symbols. 8-bit RS symbols are particularly popular, being used in the Compact Disc. Non-binary convolutional codes do exist (the "dual-k") codes, but it's also possible to adapt binary codes to the purpose.

By the way, it's important to remember that when FEC is discussed, Eb/N0 ratios always apply to the original user data bits and not to the encoded, redundant symbols. For example, a rate 1/2 K=32 convolu-

tional code can operate with an Eb/N0 down to about 3 dB. The Es, or energy per FEC encoded symbol is 1/2 (3dB) less, or 0 dB. For a rate 1/4 code the Es/N0 is 6dB less than the Eb/N0, and so on.

When FEC is layered on top of an ideal coherent modulation scheme like BPSK, lowering the code rate always improves the coding "gain" (i.e., it decreases the Eb/N0 needed to attain a certain bit error rate). But this is no longer true when FEC is combined with M-ary noncoherent demodulation. Below a certain rate, the Eb/N0 requirement actually increases again.

[Note: see diagram showing Eb/N0 vs code rate for AWGN channel]

Why is this so? A coherent demodulator is a linear device. Postprocessing of a noisy signal can improve it by coherently combining very small signal components that are still spread over time or frequency. But a noncoherent demodulator is a nonlinear device, and it has a threshold effect much like that of an FM discriminator. (We can actually think of M-ary FSK as FM with M discrete quantized steps). Above threshold, the SNR in the filter bandwidth is high enough that the accumulated signal energy swamps the noise. But below threshold, the output SNR decreases more rapidly than the input SNR.

So as we lower the FEC code rate while keeping the user data rate constant, the symbol rate coming out of the encoder increases. As the encoder symbol rate increases, we need to increase the signalling rate on the channel (the encoded FEC symbols become the data bits to the M-ary modulation scheme). And because we're spreading our fixed transmitter energy out over many more channel symbols, the energy per symbol must decrease. If it decreases below threshold, the demodulator losses increase faster than the coding gain, and overall performance suffers.

The optimum code rate depends on M and whether there is fading. On a nonfading channel, the optimum code rate is about 1/2 for a very wide range of M. On a fading channel, however, the optimum code rate is much lower, typically ranging from 1/10 to 1/3 depending on M.

[Note: see diagram showing Eb/N0 vs code rate for Rayleigh channel]

Why is this? First of all, we must understand that Eb/N0 ratios given for a fading channel refer to averages. Sometimes the signal is much greater than average, and sometimes it's much worse. When it's much worse, we pretty much have to write off our energy investment because it's so far below threshold that even a slower channel symbol rate wouldn't help. But when the signal peaks well above average, we also waste energy because each symbol carries much more energy than is necessary for reliable demodulation.

By sending at a higher channel symbol rate, we can take better advantage of these peaks when they occur. But we don't want to go too fast lest we almost never get peaks that are strong enough. So the optimum code rate is higher than for a nonfading channel.

g. Diversity:

To paraphrase a real estate agent, the three most important things in the design of a communication system for a fading channel are diversity, diversity and diversity. This is simply another way of saying that you shouldn't put all your bit "eggs" in one basket. You should spread them over as many physical dimensions as possible: in frequency, in time and in space. In so doing, you exploit the same "law of large numbers" that enables insurance companies and casinos to, on average, make money. The basic idea is that while one frequency channel, time slot or physical path may be in a fade, another may be at its peak or at least somewhere in between. With diversity you get something more like the channel's average performance all the time.

The really remarkable thing is just how close in practice it's possible to come to this ideal. With the right coding and modulation and enough bandwidth, it is fairly straightforward to achieve an average operating

Eb/N0 of only 6-7 dB on a Rayleigh fading channel. The same system might require only 3-4 dB on a nonfading channel, a "fading penalty" of only 3 dB.

Using FEC to add redundant symbols is one particularly effective way to exploit diversity in the time domain. It can also add diversity in the frequency domain if the value M is sufficiently large, which essentially constitutes spread spectrum. In fact, diversity is so powerful that the very simplest "FEC scheme" -- simple repetition of the user data -- can provide dramatic gains if the "diversity order" is high enough. (I'm sure EMEers know this instinctively, given their emphasis on repetition). In practice, though, the best systems use true FEC because the improvements come more quickly.

That the optimum FEC code rate for a fading channel is lower than for a nonfading channel is another example of diversity in action -- spreading the energy budget for each user data bit as thinly as possible in time. In the ideal signal, every instant in a transmission would be a function of every user data bit in the message. As long as you get a certain minimum fraction of the total transmitted energy in a message, you could decode it; the precise fading pattern wouldn't matter. We can come surprisingly close to this ideal.

Obviously, if we could tell when (or where) the channel is going to be at its peak and send at high speed just at those times we could avoid wasting a lot of energy on deep fades. But this is much easier said than done. Even with the slow fading seen on 2m EME, the long round trip delay to the moon and back means that by the time we detected a peak and let the other station know about it, it would probably be gone. So we have no real choice but to hedge our bets and diversify.

h.Interleaving:

Certain FEC codes are limited in their ability to "spread out" (i.e., diversify) the effect of each user data bit widely enough in time. A long deep fade may take out all of the coded symbols affected by a single data bit, causing it to be lost. This is particularly true for convolutional coding, which otherwise has the significant advantage of being adaptable to "soft decision" decoding. (Long block codes like the Reed-Solomon code don't have this problem, but they are not readily adapted to soft decision decoding).

It turns out that a very simple trick can nearly solve this problem. Instead of sending the FEC encoded symbols in the exact order they are sent, you can scramble them in time before transmission and put them back in the right order at the receiver before decoding. A long fade (burst error) is then transformed into widely scattered short errors that the code can easily handle.

The main parameter for an interleaver is the "span", i.e., the range over which adjacent symbols are scattered in time. It's important that the span be considerably larger than the coherence time to maximize the benefit. Memory being cheap, the only real limit in practice on interleaving span is the maximum delay the user will allow. In a packet-like environment, interleaving can and probably should be over the entire packet.

[Note: see diagram of typical block interleaver]

7. Summary and Recommendations for EME:

So putting all this together, what kind of modulation and coding schemes can we recommend for the EME channel?

The overall structure is now clear: efficient source coding of the EME message, as discussed earlier, a forward error correction encoder, an interleaver and a M-ary FSK modulator feeding the transmitter. All these steps up to the generation of the transmitted audio waveform can be done in software on a PC driving a standard sound card.

Eb/N0 of only 6-7 dB on a Rayleigh fading channel. The same system might require only 3-4 dB on a nonfading channel, a "fading penalty" of only 3 dB.

Using FEC to add redundant symbols is one particularly effective way to exploit diversity in the time domain. It can also add diversity in the frequency domain if the value M is sufficiently large, which essentially constitutes spread spectrum. In fact, diversity is so powerful that the very simplest "FEC scheme" -- simple repetition of the user data -- can provide dramatic gains if the "diversity order" is high enough. (I'm sure EMEers know this instinctively, given their emphasis on repetition). In practice, though, the best systems use true FEC because the improvements come more quickly.

That the optimum FEC code rate for a fading channel is lower than for a nonfading channel is another example of diversity in action -- spreading the energy budget for each user data bit as thinly as possible in time. In the ideal signal, every instant in a transmission would be a function of every user data bit in the message. As long as you get a certain minimum fraction of the total transmitted energy in a message, you could decode it; the precise fading pattern wouldn't matter. We can come surprisingly close to this ideal.

Obviously, if we could tell when (or where) the channel is going to be at its peak and send at high speed just at those times we could avoid wasting a lot of energy on deep fades. But this is much easier said than done. Even with the slow fading seen on 2m EME, the long round trip delay to the moon and back means that by the time we detected a peak and let the other station know about it, it would probably be gone. So we have no real choice but to hedge our bets and diversify.

h.Interleaving:

Certain FEC codes are limited in their ability to "spread out" (i.e., diversify) the effect of each user data bit widely enough in time. A long deep fade may take out all of the coded symbols affected by a single data bit, causing it to be lost. This is particularly true for convolutional coding, which otherwise has the significant advantage of being adaptable to "soft decision" decoding. (Long block codes like the Reed-Solomon code don't have this problem, but they are not readily adapted to soft decision decoding).

It turns out that a very simple trick can nearly solve this problem. Instead of sending the FEC encoded symbols in the exact order they are sent, you can scramble them in time before transmission and put them back in the right order at the receiver before decoding. A long fade (burst error) is then transformed into widely scattered short errors that the code can easily handle.

The main parameter for an interleaver is the "span", i.e., the range over which adjacent symbols are scattered in time. It's important that the span be considerably larger than the coherence time to maximize the benefit. Memory being cheap, the only real limit in practice on interleaving span is the maximum delay the user will allow. In a packet-like environment, interleaving can and probably should be over the entire packet.

[Note: see diagram of typical block interleaver]

7. Summary and Recommendations for EME:

So putting all this together, what kind of modulation and coding schemes can we recommend for the EME channel?

The overall structure is now clear: efficient source coding of the EME message, as discussed earlier, a forward error correction encoder, an interleaver and a M-ary FSK modulator feeding the transmitter. All these steps up to the generation of the transmitted audio waveform can be done in software on a PC driving a standard sound card.

It is interesting to consider scaling this modulation scheme up to the capabilities of full scale EME stations. The link calculations performed for the EME station described at the beginning of this paper show that a continuous user data rate of 300bps should be possible. However there are scaling problems caused by the immutable link parameters such as coherence time and bandwidth, plus somewhat more changeable parameters such as transceiver bandwidth.

Since a better link could support a faster MFSK symbol rate, we could move away from the limit imposed by coherence time. In fact, we could use this margin to increase M and further lower our Eb/N_0 requirements. But even if we keep M=64, scaling this scheme up to 300 bits/sec would need almost 64KHz of bandwidth, well beyond the range of most transceivers and PC sound cards. To stay within these limits we'd have to decrease M and sacrifice Eb/N_0 performance somewhat. Also, with faster symbol times we'd need more precise symbol timing, though with GPS and accurate real-time computer calculations of moon and station position this might not be hard to do.

A more fundamental problem is the inherent EME delay spread of several milliseconds. If the symbol interval is shortened to where the delay spread becomes an appreciable fraction, intersymbol interference appears. The best way around this is to frequency hop or chirp from symbol to symbol to allow time for the limb reflections on a given frequency to die down before it is used again -- and this means still more total RF bandwidth.

So it seems pretty clear that the potential of a conventional (large) EME station can be fully realized only with a true spread-spectrum signal.

On the other hand, scaling this technique to even smaller stations runs into more fundamental problems. The big barrier is the channel coherence time; with our parameters we are already perhaps too close to it. There would be no choice but to limit symbol times to less than desirable values. We could keep M=64 and reduce the FEC code rate to that which the link can support, and we could also increase M. But in either case, the FEC code rate would probably have to be well below the optimum for the corresponding value of M on the Rayleigh fading channel. The FEC decoder would then be in the position of noncoherently combining symbol energy in a less than optimum manner, raising the overall Eb/N_0 requirements and further lowering the attainable user data rate. The system would work, but not nearly as efficiently. And it would certainly tax the patience of most operators since our scheme necessarily gives no real feedback to the operator until after an entire transmission has been transmitted and received.

July 24, 1996

Tom Clark, W3IWI
clark@tomcat.gsfc.nasa.gov
-or-
w3iwi@amsat.org

Phil Karn, KA9Q
karn@qualcomm.com
-or-
ka9q@amsat.org